

CLAIMS

What is claimed is:

- 1 1. A method, comprising:
2 dynamically invoking a speculative thread from a main thread in a processor;
3 and
4 executing instructions comprising the speculative thread.
- 1 2. The method of claim 1, further comprising attempting to bind a hardware thread
2 context for the speculative thread.
- 1 3. The method of claim 2, further comprising determining whether the attempt to bind
2 the hardware thread context for the speculative thread was a success or a failure.
- 1 4. The method of claim 3, further comprising raising an exception and branching to a
2 recovery handler when it is determined that the attempt to bind a hardware thread context for
3 the speculative thread was a success.
- 1 5. The method of claim 3, further comprising treating the attempt to bind a hardware
2 thread context for the speculative thread as a no operation instruction when it is determined
3 that the attempt to bind a hardware thread context for the speculative thread was a failure.
4
- 1 6. The method of claim 1, further comprising transferring live-in values for pre-
2 computation slices to hardware thread context register files.
- 1 7. The method of claim 6, further comprising allocating a portion of memory for the
2 speculative thread and dedicating the allocated a portion of on-chip memory buffer space as
3 an intermediate buffer to pass live-in values from the main thread to the speculative thread.

1 8. The method of claim 6, further comprising:
2 identifying at least one load of interest;
3 constructing pre-computation slices for each load of interest, wherein the pre-
4 computation slice comprises a speculative thread that pre-computes an address accessed by a
5 load of interest and pre-fetching for the load of interest using the pre-computed address; and
6 establishing triggers to invoke the speculative thread.

1 9. A method in a processor, comprising:
2 dynamically invoking from a first speculative thread a second speculative
3 thread, wherein the first speculative thread is dynamically invoked from a main thread; and
4 executing instructions comprising the first and second speculative threads.

1 10. The method of claim 9, further comprising attempting to bind a hardware thread
2 context for the speculative thread.

1 11. The method of claim 9, further comprising detecting a trigger to invoke the second
2 speculative thread, storing second speculative thread live-in values to a buffer, and executing
3 instructions in the second speculative thread.

1 12. The method of claim 10 wherein each speculative thread includes a pre-computation
2 slice, the method further comprising:

3 allocating an entry in a queue for a copy of at least one pre-computation slice
4 when it is determined that a hardware context is unavailable for the copy of the pre-
5 computation slice; and

6 placing the pre-computation slice in the queue until a hardware context is
7 available.

1 13. A processor, comprising:
2 a first hardware context to store a main software thread;
3 a second hardware context to store a speculative software thread;
4 logic coupled between the first and second hardware contexts and a machine-
5 readable medium having machine-readable instructions stored thereon to instruct a processor
6 to bind the speculative software thread to the second hardware context and to transfer live-in
7 values from main software thread to the speculative software thread; and

1 14. The processor of claim 13 wherein the logic to copy live-in values from first hardware
2 context to the second hardware context includes flash-copy hardware or portion of memory
3 buffer space.

1 15. A processor, comprising:
2 a first hardware context to store a main thread;
3 a second hardware context and a third hardware context to bind to a first
4 speculative thread and a second speculative thread, respectively, the main thread to
5 dynamically invoke the first speculative thread and the first speculative thread to dynamically
6 invoke the second speculative thread; and logic coupled between the first, second, and third
7 hardware contexts, and a processor-readable medium having processor-readable instructions
8 stored thereon to instruct the processor, to bind the first and second speculative threads to the
9 second and third hardware contexts, respectively, and to transfer live-in values from main
10 thread to the first speculative thread and from the first speculative thread to the second
11 speculative thread, wherein the processor-readable medium includes at least one instruction
12 to instruct the processor to trigger the invocation of the first and second speculative threads.

1 16. The processor of claim 15, further comprising a pending slice queue having entries to
2 allocate a copy of at least one pre-computation slice when it is determined that a hardware
3 context is unavailable for the copy of the pre-computation slice and to hold the pre-
4 computation slice until a hardware context is available.
5

1 17. The processor of claim 15, further comprising an outstanding pre-computation slice
2 counter to track for a set of loads of interest the number speculative threads that have been
3 spawned relative to the number of instances of any load of interest that have not yet been
4 retired by the main thread and to decrement en the main thread retires the corresponding load
5 of interest.

1 18. The processor of claim 15, further comprising an outstanding pre-computation slice
2 counter to track for a set of loads of interest the number speculative threads that have been
3 spawned relative to the number of instances of any load of interest that have not yet been
4 retired by the main thread and to decrement en the main thread retires the corresponding load
5 of interest, to increment the counter when the corresponding pre-computation slice is
6 spawned, and to force any speculative thread for which the entry in the counter is in a
7 predetermined state to wait in the queue until the entry in the counter becomes a second
8 predetermined state.
1

1 19. A machine-readable medium having machine-readable instructions stored thereon to
2 instruct a processor to:
3 dynamically invoke a speculative thread from a main thread; and
4 execute instructions comprising the speculative thread.

1 20. The machine-readable medium of claim 19 wherein the instructions are further to
2 instruct the processor to:

3 allocate an entry in a queue for a copy of at least one pre-computation slice
4 when it is determined that a hardware context is unavailable for the copy of the pre-
5 computation slice; and
6 place the pre-computation slice in the queue until a hardware context is
7 available.
8

1 21. The machine-readable medium of claim 19 wherein the instructions are further to
2 instruct the processor to:
3 bind a hardware context for the speculative thread;
4 transfer live-in values from the main thread to the speculative thread; and
5 load live-in values in the hardware context for the speculative thread.

1 22. The machine-readable medium of claim 19 wherein the instructions are further to
2 instruct the processor to:
3 identify a set of loads of interest;
4 construct pre-computation slices for each delinquent load of interest, wherein
5 the pre-computation slice comprises a speculative thread that pre-computes an address
6 accessed by a load of interest; and establish triggers to invoke the speculative thread.

1 23. The machine-readable medium of claim 22 wherein the instructions are further to
2 instruct the processor to:
3 allocate an entry in a queue for a copy of at least one pre-computation slice
4 when it is determined that a hardware context is unavailable for the copy of the pre-
5 computation slice; and
6 place the pre-computation slice in the queue until a hardware context is
7 available.
8

1 24. The machine-readable medium of claim 23 wherein the instructions are further to
2 instruct the processor to:

3 track for a subset of the set of loads of interest the number speculative threads
4 that have been spawned relative to the number of instances of any load of interest that have
5 not yet been retired by the main thread; and

6 decrement a counter when the main thread retires the corresponding load of
7 interest.

1 25. The machine-readable medium of claim 23 wherein the instructions are further to
2 instruct the processor to:

3 track for a subset of the set of loads of interest the number speculative threads
4 that have been spawned relative to the number of instances of any load of interest that have
5 not yet been retired by the main thread; and

6 increment a counter when the corresponding pre-computation slice is
7 spawned; and

8 forcing any speculative thread for which the entry in the counter is in a
9 predetermined state to wait in the queue until the entry in the counter becomes a second
10 predetermined state.

1

1 26. A machine-readable medium having machine-readable instructions stored thereon to
2 instruct a processor to:

3 dynamically invoke from a first speculative thread a second speculative
4 thread, the first speculative thread to be dynamically invoked from a main thread; and

5 execute instructions comprising the first and second speculative threads
6 invoke a speculative thread from a main thread.

1 27. The machine-readable medium of claim 26 wherein the instructions are further to
2 instruct the processor to:

3 bind a first and second hardware contexts for the first and second speculative
4 threads, respectively; and

5 transfer live-in values from the main thread to the first hardware context and
6 from the first hardware context to the second hardware context.

1 28. The machine-readable medium of claim 27 wherein the instructions are further to
2 instruct the processor to:

3 identify a set of loads of interest;

4 construct pre-computation slices for each delinquent load of interest, wherein
5 the pre-computation slice comprises a speculative thread that pre-computes an address
6 accessed by a load of interest; and

7 establish triggers to invoke the speculative thread.

1 29. The machine-readable medium of claim 28 wherein the instructions are further to
2 instruct the processor to:

3 allocate an entry in a queue for a copy of at least one pre-computation slice
4 when it is determined that a hardware context is unavailable for the copy of the pre-
5 computation slice; and

6 place the pre-computation slice in the queue until a hardware context is
7 available.

1 30. The machine-readable medium of claim 28 wherein the instructions are further to
2 instruct the processor to:

3 track for a subset of the set of loads of interest the number speculative threads
4 that have been spawned relative to the number of instances of any load of interest that have
5 not yet been retired by the main thread; and
6 decrement a counter when the main thread retires the corresponding load of
7 interest.

1 31. The machine-readable medium of claim 28 wherein the instructions are further to
2 instruct the processor to:

3 track for a subset of the set of loads of interest the number speculative threads
4 that have been spawned relative to the number of instances of any load of interest that have
5 not yet been retired by the main thread; and

6 increment a counter when the corresponding pre-computation slice is
7 spawned; and

8 forcing any speculative thread for which the entry in the counter is in a
9 predetermined state to wait in the queue until the entry in the counter becomes a second
10 predetermined state.